

REMARKS

Applicants thank the Examiner for total consideration given the present application. Claims 1-19 and 22 were pending prior to the Office Action. Claims 1-5, 14-19, and 22 have been canceled and claims 23-30 have been added through this Reply. Therefore, claims 6-13 and 23-30 are currently pending. Claims 6 and 23 are independent. Applicants appreciate that the previous arguments filed on October 23, 2007 were found persuasive. However, claims 1-19 and 22 now stand rejected under a new ground(s) of rejection. Applicants respectfully request reconsideration of the rejected claims in light of the amendment and remarks presented herein, and earnestly seek timely allowance of all pending claims.

35 U.S.C. § 112, first and second paragraph- Rejections

Claims 1-5 are rejected under 35 U.S.C. § 112, first paragraph, as allegedly failing to comply with the written description requirement. The Examiner alleges that these claims contain subject matter which was not described in the specification in such a way as to reasonably convey to one skilled in the relevant art that the inventors, at the time the application was filed, had possession of the claimed invention. Although Applicants do not necessarily agree with the Examiner regarding this issue, claims 1-5 have been canceled through the Reply rendering the rejection of these claims as moot.

Claims 1-5 are also rejected under 35 U.S.C. § 112, second paragraph, as allegedly being indefinite for failing to particularly point out and distinctly claim the subject matter which Applicants regard as the invention. Although Applicants do not necessarily agree with the Examiner that claims 1-5 are indefinite, these claims have been canceled through this Reply rendering the rejection of these claims as moot.

35 U.S.C. § 101 - Rejection

Claims 1-5 are rejected under 35 U.S.C. § 101 because the claimed invention is directed to non-statutory subject matter. Although Applicants do not necessarily agree with the Examiner that claims 1-5 are non-statutory, these claims have been canceled through this Reply rendering the rejection of these claims as moot.

35 U.S.C. § 103 - Rejection

A. Claims 1-5 and 22 stand rejected under 35 U.S.C. § 103(a) as allegedly being unpatentable over Bahrs et al. (U.S. Publication No. 2003/0097650)[hereinafter "Bahrs"], in view of Bischof et al. (U.S. Publication No. 2004/0041827)[hereinafter "Bischof"]. These claims have been canceled through this Reply, rendering the rejection of these claims as moot.

B. Claims 6-19 stand rejected under 35 U.S.C. § 103(a) as allegedly being unpatentable over Bahrs. Applicants respectfully traverse this rejection. Claims 14-19 have been canceled through this Reply rendering the rejection of claims 14-19 as moot.

For a Section 103 rejection to be proper, a *prima facie* case of obviousness must be established. *See M.P.E.P. 2142*. One requirement to establish *prima facie* case of obviousness is that the prior art references, when combined, must teach or suggest all claim limitations. *See M.P.E.P. 2142; M.P.E.P. 706.02(j)*. Thus, if the cited references fail to teach or suggest one or more elements, then the rejection is improper and must be withdrawn.

In this instance, it is respectfully submitted that Bahrs does not teach or suggest all claim limitations. For example, amended independent claim 6 recites, *inter alia*,

“extracting parameter value combinations from a data file formatted with a markup language listed in a parameter order to implement data of an external table associated with a first test case; wherein extracting parameter value combinations includes extracting a first set of parameter values and listing the first set of parameter values in an order such that each value in said first set of parameter values is positioned in the same order as the corresponding parameter listed in the parameter order;

transmitting the parameter value combinations and the first set of parameter values to a software module test engine, wherein the parameter value combinations and the first set of parameter values are identified with the first test case; . . .

changing the data file to implement data of the external table associated with a second test case for testing the software module, and extracting a second set of parameter values and listing the second set of parameter values in an order such that each value in said second set of parameter values is positioned in the same order as the corresponding parameter listed in the parameter order, wherein the parameter value combinations and the second set of parameter values are identified with the second test case; . . .” *Emphasis added*.

It is respectfully submitted that Bahrs fails to teach the above-identified claim features. Further, *assuming arguendo*, even if Bischof is combined with Bahrs, the combined invention

would fail to teach or suggest the above-identified claim features as recited in independent claim 6.

Bahrs fails to teach the above-identified claim features of claim 6

Bahrs merely discloses a conventional method for testing software by loading a software component onto a data processing system and reading from a configuration data structure for a test case. Then the software component is executed using the test case in which an actual result is generated. The actual result is compared with an expected result to determine whether an error has occurred. (See abstract.)

In Fig. 7, Bahrs discloses a flowchart of a process used for executing the test case implemented in a test harness, such as test harness 502 shown in Fig. 5. As taught by Bahrs, this process begins by loading a configuration file (step 700) located in the data structure, such as an XML file. Objects are initialized using the configuration file (step 702) and then the test mediator is initialized (step 704). The test mediator is executed (step 706) with the process terminating thereafter. Bahrs continues to disclose that when the test mediator is tested or invoked by the test harness on the test case, the test mediator will execute the test case(s). In these examples, more than one test case may be loaded and tested by the process. Additionally, the test harness will control the number of iterations required. For example, if five iterations are requested, then the test mediator is created or invoked five times by the test harness. Alternatively, the test harness may create a single test mediator and run the test five times. The control of iterations, as well as the throttling of the test, occurs within step 706 in these examples. In Fig. 8, Bahrs illustrates an example of attributes associated with the test harness. (See paragraphs [0050] to [0052].)

With reference to Fig. 13, Bahrs discloses a flowchart of a process used for comparing test results in which the process begins by parsing the actual results (step 1300). These actual results are the results returned from the test component. Bahrs further discloses that the parsing of the data that is to be compared may be identified by information in the configuration file. Then the data from the actual results is converted into a first hash table (step 1302). The expected

results are parsed (step 1304). The description of this data is also described in the configuration file. The data from the expected results is converted into a second hash table (step 1306); the hash tables are then compared (step 1308). Next, a determination is made as to whether there is a match between the values in the first and second hash table (step 1310). If there is a match between the first and second hash table, no error is returned (step 1312) and the process terminates thereafter. With reference again to step 1310, if there is not a match between the first and second hash table, an error is returned (step 1314) with the process terminating thereafter. (See paragraphs [0068] and [0069].)

Bahrs is distinguished from the claimed invention in that nowhere does Bahrs teach or suggest the steps of “extracting parameter value combinations from a data file formatted with a markup language listed in a parameter order to implement data of an external table associated with a first test case; wherein extracting parameter value combinations includes extracting a first set of parameter values and listing the first set of parameter values in an order such that each value in said first set of parameter values is positioned in the same order as the corresponding parameter listed in the parameter order . . . wherein the parameter value combinations and the first set of parameter values are identified with the first test case; . . . changing the data file to implement data of the external table associated with a second test case for testing the software module, and extracting a second set of parameter values and listing the second set of parameter values in an order such that each value in said second set of parameter values is positioned in the same order as the corresponding parameter listed in the parameter order” as recited in independent claim 6.

The Examiner acknowledges that Bahrs does not explicitly disclose that “the first set and second set of extracted parameter values are listed in an order such that each value in said first set of parameter values is positioned in the same order as their respective and corresponding parameter listed in the . . . parameter order” (see page 6, 4th full paragraph of the Office Action). However, the Examiner alleges that markup language for creating tagged components or variables and associating tagged type or value thereof to those variables was a known concept at the time the invention was made (see page 6, 4th full paragraph of the Office Action). It is

respectfully submitted that Applicants find *no correlation* between the above statement of the Examiner and the claimed invention. The claimed invention is directed to *extracting a first set of parameter values* and *listing the first set of parameter values in an order* such that each value in said first set of parameter values is positioned *in the same order* as the corresponding parameter listed in the parameter order. The Examiner has *failed to provide any evidence* that “a markup language for creating tagged components or variables and associated tagged type or value thereof to those variables” would establish a *prima facie* obviousness of the claimed invention. The “configuration file” (shown in page 6, paragraph [0069]) as relied upon by the Examiner has *no* indication that a same order exists for the parameters and their corresponding values even if a plurality of test cases are created based on a section of the configuration file.

In addition, it is respectfully submitted that nowhere does Bahrs teach or suggest that the *parameter value combinations and the first set of parameter values are identified with the first test case*. Indeed, Bahrs is completely silent on any teaching of “parameter value combinations”, let alone identifying the parameter value combinations with a first test case or second test case as recited in claim 6.

Therefore, for at least these reasons, it is respectfully submitted that Bahrs is distinguished from the claimed invention as recited in claim 6.

Bischof fails to teach the above-identified claim features of claim 6

Bischof discloses a test configuration 140 to test particular functionality of an application program. This test configuration can encapsulate data and scripts to test one or more particular program functions in one or more user interface environments. For example, the test configuration can include test data 145, system data 150, and a multi-platform (e.g., non-client specific) test script object 155. The test data 145 and system data 150 are generic--that is, they are not specific for any particularly functionality--and so it will generally be more convenient to implement test configuration 140 as a persistent object containing a link to this data, rather than

the data itself. The script object 155, on the other hand, is specific for one or more particular program functions. (See page 3, paragraph [0025].)

More specifically, in Fig. 6, Bischof discloses a process 600 for using a test script to test application functionality in different user interface environments. The process begins by receiving, for example, from a test manager, an identification of a test script and an identification of a target system, that is, of a computer system and user interface environment to be tested (step 610). The process obtains the specified test script (step 620), which includes one or more commands calling a command interface or interfaces specifying actions to be performed to test the operation of the application and user interface in a user interface environment. The test script can also include commands directly invoking application functions (e.g., through corresponding command interfaces). After executing the specified test script (step 630), the process identifies first or a subsequent user element, property/method, and parameter(s), if required, and provides input parameter values to GUI field (steps 640, 650, 680).

Similar to Bahrs, Bischof is also distinguished from the claimed invention in that nowhere does Bischof teach or suggest the steps of “extracting parameter value combinations from a data file formatted with a markup language listed in a parameter order to implement data of an external table associated with a first test case; wherein extracting parameter value combinations includes extracting a first set of parameter values and listing the first set of parameter values in an order such that each value in said first set of parameter values is positioned in the same order as the corresponding parameter listed in the parameter order . . . wherein the parameter value combinations and the first set of parameter values are identified with the first test case; . . . changing the data file to implement data of the external table associated with a second test case for testing the software module, and extracting a second set of parameter values and listing the second set of parameter values in an order such that each value in said second set of parameter values is positioned in the same order as the corresponding parameter listed in the parameter order” as recited in independent claim 6.

The Examiner relies on paragraphs [0041]-[0043], Fig. 6, and listing 1 (shown in page 5) and jumps to the conclusion that “Bischof has disclosed extracted parameter values such that each value in said second set of parameter values (each of the parameter value) is positioned in the same layout order as the corresponding parameter has been listed in the source parameter tag section, i.e. the first section of test parameters. Based on the more test cases being derived from a plurality of XML configuration files, it would have been obvious for one skilled in the art at the time the invention was made to implement the extraction of parameters in Bahrs XML file such that extracted parameters are listed in an order such that each value in said first set of parameter values is positioned in the same order as the corresponding parameter listed in the parameter order; and associating these extracted parameters and values for one first among more test cases.” (See page 7, lines 5-22 of the Office Action.)

It is respectfully submitted that the Examiner’s interpretation of the relied upon sections of Bischof is totally erroneous. Bischof, in the relied upon sections, merely discloses an example of an “abstract representation” and how it can be generated as an XML document. Such generation of an “abstract representation” fails to teach or suggest the specifics of “extracting parameter value combinations”; “listing the first set of parameter values in an order such that each value in said first set of parameter values is positioned in the same order as the corresponding parameter listed in the parameter order” and “extracting a second set of parameter values and listing the second set of parameter values in an order such that each value in said second set of parameter values is positioned in the same order as the corresponding parameter listed in the parameter order” as recited in amended independent claim 6.

Therefore, for at least these reasons, it is respectfully submitted that Bischof is distinguished from the claimed invention as recited in claim 6.

Accordingly, it is respectfully submitted that independent claim 6 is allowable over Bahrs and Bischof. Claims 7-13 are at least allowable by virtue of their dependency on allowable claim 6.

New Claims

New claims 23-30 are directed to a computer-readable medium having computer executable program for performing the steps of claims 6-13 respectively. Thus, these claims are at least allowable for the same reasons stated above with respect to claims 6-13.

Conclusion

In view of the above remarks, it is believed that all pending claims are allowable.

Should there be any outstanding matters that need to be resolved in the present application, the Examiner is respectfully requested to contact Ali M. Imam Reg. No. 58,755 at the telephone number of the undersigned below, to conduct an interview in an effort to expedite prosecution in connection with the present application.

If necessary, the Commissioner is hereby authorized in this, concurrent, and future replies to charge payment or credit any overpayment to Deposit Account No. 02-2448 for any additional fees required under 37.C.F.R. §§1.16 or 1.14; particularly, extension of time fees.

Dated: March 12, 2008

Respectfully submitted,

By  #58,755
Michael K. Mutter
Registration No.: 29,680
BIRCH, STEWART, KOLASCH & BIRCH, LLP
8110 Gatehouse Road
Suite 100 East
P.O. Box 747
Falls Church, Virginia 22040-0747
(703) 205-8000
Attorney for Applicant